



(19)

(11) Publication number: 2000155673 A

Generated Document

PATENT ABSTRACTS OF JAPAN

(21) Application number: 10343632

(51) Intl. Cl.: G06F 9/06

(22) Application date: 18.11.98

(30) Priority:

(43) Date of application
publication: 06.06.00(84) Designated
contracting states:

(71) Applicant: NEC IC MICROCOMPUT SYST LTD

(72) Inventor: ANDO YOSHINARI
SHAMOTO EIJI

(74) Representative:

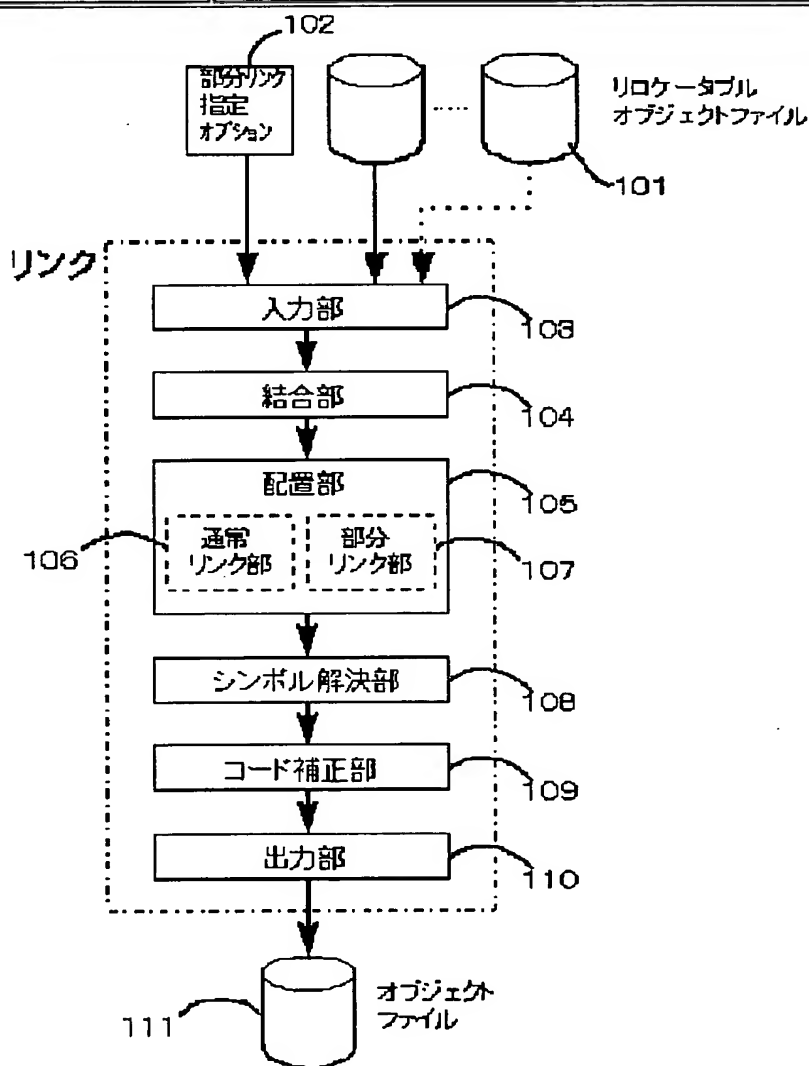
(54) LINK DEVICE AND LINK
METHOD

(57) Abstract:

PROBLEM TO BE SOLVED: To shorten a time required for link processing by solving a locatable symbol value according to an arrangement address decided by a partial link processing, and correcting an instruction code.

SOLUTION: When an inputting part 103 inputs each kind of file 101 and 102, a connecting part 104 gathers the same name segments in a file. An arranging part 105 judges the designation of partial link processing according to designation to the input part 103 at the time of starting a link device, and a normal link part 106 operates general whole link, and a partial link part 107 operates partial link processing for inserting a dead space into the end of a segment to be arranged. A symbol settling part 108 solves a locatable symbol value according to an arrangement address decided by an arranging part 105. A code correcting part 109 corrects an instruction code by using the solved symbol value for an instruction to refer to the symbol value. An output part 110 generates an object file 111 including the solved code.

COPYRIGHT: (C)2000,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-155673
(P2000-155673A)

(43) 公開日 平成12年6月6日(2000.6.6)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
G 0 6 F 9/06	4 1 0	G 0 6 F 9/06	4 1 0 E 5 B 0 7 6

審査請求 有 請求項の数12 F D (全 24 頁)

(21) 出願番号 特願平10-343632
(22) 出願日 平成10年11月18日(1998.11.18)

(71) 出願人 000232036
日本電気アイシーマイコンシステム株式会
社
神奈川県川崎市中原区小杉町1丁目403番
53
(72) 発明者 安藤 喜成
神奈川県川崎市中原区小杉町一丁目403番
53 日本電気アイシーマイコンシステム株
式会社内
(74) 代理人 100097113
弁理士 堀 城之

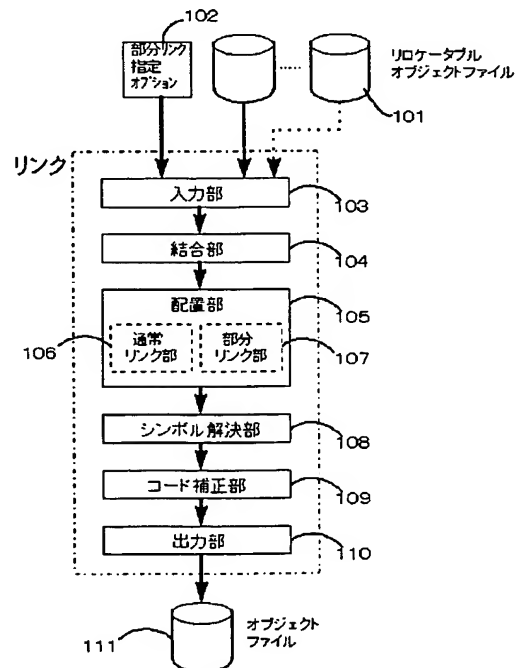
最終頁に続く

(54) 【発明の名称】 リンク装置及びリンク方法

(57) 【要約】

【課題】 本発明は、リンク処理に要する時間を低減で
きるリンク及びリンク方法を提供することを課題とす
る。

【解決手段】 複数のオブジェクト・モジュールをリン
クして1つの実行可能プログラムを生成するリンクであ
って、オブジェクト・モジュールのプログラムの構成要
素であるセグメントの中で配置対象となるセグメントを
検出する手段と、検出された配置対象セグメント毎に、
システム側で固定されて割り当てられた所定サイズの空
き領域を割り当てる手段と、オブジェクト・モジュール
のプログラムに変更があった配置対象セグメントのみを
再度リンクする部分リンク処理を実行する手段と、部分
リンク処理を実行する際、配置対象セグメント毎にシス
テム側で固定されて割り当てられた所定サイズの空き領
域を対応する配置対象セグメントの終わりに各々付加す
る手段とを有する。



【特許請求の範囲】

【請求項1】 複数のオブジェクト・モジュールをリンクして1つの実行可能プログラムを生成するリンクであって、

オブジェクト・モジュールのプログラムの構成要素であるセグメントの中で配置対象となるセグメントを検出する手段と、

当該検出された配置対象セグメント毎に、システム側で固定されて割り当てられた所定サイズの空き領域を割り当てる手段と、

前記オブジェクト・モジュールのプログラムに変更があった前記配置対象セグメントのみを再度リンクする部分リンク処理を実行する手段と、

前記部分リンク処理を実行する際、前記配置対象セグメント毎にシステム側で固定されて割り当てられた前記所定サイズの空き領域を当該対応する配置対象セグメントの終わりに各々付加する手段とを有することを特徴とするリンク。

【請求項2】 前記配置対象セグメントがある場合、現在の前記部分リンク処理が1回目の部分リンクか、または前記部分リンク処理での全リンクかを判断する手段と、

前記1回目の部分リンクまたは前記部分リンク処理での全リンクの場合、前記配置対象セグメントを配置する手段と、

当該配置した配置対象セグメントの終わりに挿入する前記空き領域のサイズをシステム側で固定値に決定する手段と、

当該固定の大きさの空き領域を当該対応する配置対象セグメントの終わりに各々挿入する手段とを有することを特徴とする請求項1に記載のリンク。

【請求項3】 複数のオブジェクト・モジュールをリンクして1つの実行可能プログラムを生成するリンクであって、

オブジェクト・モジュールのプログラムの構成要素であるセグメントの中で配置対象となるセグメントを検出する手段と、

当該検出された配置対象セグメント毎に、ユーザ側が任意に割り当てた所定サイズの空き領域を割り当てる手段と、

前記オブジェクト・モジュールのプログラムに変更があった前記配置対象セグメントのみを再度リンクする部分リンク処理を実行する手段と、

前記部分リンク処理を実行する際、前記配置対象セグメント毎にユーザ側が任意に割り当てた前記所定サイズの空き領域を当該対応する配置対象セグメントの終わりに各々付加する手段とを有することを特徴とするリンク。

【請求項4】 前記配置対象セグメントがある場合、現在の前記部分リンク処理が1回目の部分リンクか、または前記部分リンク処理での全リンクかを判断する手段

と、

前記1回目の部分リンクまたは前記部分リンク処理での全リンクの場合、前記配置対象セグメントを配置する手段と、

当該配置した配置対象セグメントの終わりに挿入する前記空き領域のサイズをユーザ側で任意の大きさに決定する手段と、

当該任意の大きさの空き領域を当該対応する配置対象セグメントの終わりに各々挿入する手段とを有することを特徴とする請求項3に記載のリンク。

【請求項5】 前記部分リンク処理の指示を入力する入力部と、

前記配置対象セグメントの配置アドレスを決定する配置部と、

前記配置部で決定した前記配置アドレスに基づいてリロケータブルなシンボルのシンボル値を解決するシンボル解決部と、

前記リロケータブルなシンボルを参照している命令に基づいて解決した前記シンボル値を用いて命令コードの補正を行うコード補正部と、

前記解決したコードを含む出力ファイルであるオブジェクトファイルを生成する出力部とを有することを特徴とする請求項1乃至4のいずれか一項に記載のリンク。

【請求項6】 前記配置部は、

一般的なリンクである全リンクを行う通常リンク部と、前記配置対象セグメントの終わりに前記空き領域を挿入するとともに、コード修正に応じたセグメントサイズの増減に対する前記部分リンク処理を行う部分リンク部を有することを特徴とする請求項5に記載のリンク。

【請求項7】 複数のオブジェクト・モジュールをリンクして1つの実行可能プログラムを生成するリンク方法であって、

オブジェクト・モジュールのプログラムの構成要素であるセグメントの中で配置対象となるセグメントを検出する工程と、

当該検出された配置対象セグメント毎に、システム側で固定されて割り当てられた所定サイズの空き領域を割り当てる工程と、

前記オブジェクト・モジュールのプログラムに変更があった前記配置対象セグメントのみを再度リンクする部分リンク処理を実行する工程と、

前記部分リンク処理を実行する際、前記配置対象セグメント毎にシステム側で固定されて割り当てられた前記所定サイズの空き領域を当該対応する配置対象セグメントの終わりに各々付加する工程とを有することを特徴とするリンク方法。

【請求項8】 前記配置対象セグメントがある場合、現在の前記部分リンク処理が1回目の部分リンクか、または前記部分リンク処理での全リンクかを判断する工程と、

前記1回目の部分リンクまたは前記部分リンク処理での全リンクの場合、前記配置対象セグメントを配置する工程と、

当該配置した配置対象セグメントの終わりに挿入する前記空き領域のサイズをシステム側で固定値に決定する工程と、

当該固定の大きさの空き領域を当該対応する配置対象セグメントの終わりに各々挿入する工程とを有することを特徴とする請求項7に記載のリンク方法。

【請求項9】 複数のオブジェクト・モジュールをリンクして1つの実行可能プログラムを生成するリンク方法であって、

オブジェクト・モジュールのプログラムの構成要素であるセグメントの中で配置対象となるセグメントを検出する工程と、

当該検出された配置対象セグメント毎に、ユーザ側が任意に割り当てた所定サイズの空き領域を割り当てる工程と、

前記オブジェクト・モジュールのプログラムに変更があった前記配置対象セグメントのみを再度リンクする部分リンク処理を実行する工程と、

前記部分リンク処理を実行する際、前記配置対象セグメント毎にユーザ側が任意に割り当てた前記所定サイズの空き領域を当該対応する配置対象セグメントの終わりに各々付加する工程とを有することを特徴とするリンク方法。

【請求項10】 前記配置対象セグメントがある場合、現在の前記部分リンク処理が1回目の部分リンクか、または前記部分リンク処理での全リンクかを判断する工程と、

前記1回目の部分リンクまたは前記部分リンク処理での全リンクの場合、前記配置対象セグメントを配置する工程と、

当該配置した配置対象セグメントの終わりに挿入する前記空き領域のサイズをユーザ側で任意の大きさに決定する工程と、

当該任意の大きさの空き領域を当該対応する配置対象セグメントの終わりに各々挿入する工程とを有することを特徴とする請求項9に記載のリンク方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、リンク技術に関し、特に、複数のオブジェクト・モジュールを結合（リンク）して1つの実行可能プログラムを生成するリンク法及びリンク方法に関する。

【0002】

【従来の技術】近年のアプリケーションプログラム開発において、プログラム規模は増大し、より複雑なものになっている。一般的にアプリケーションプログラムのデバッグ行程では、コンパイル（アセンブル言語の場合は

アセンブル）→リンク→デバッグ→コード修正という一連の作業を繰り返し行う。アプリケーションプログラムの規模増大により、上記の各作業に要する時間が大きくなっている。

【0003】リンク処理時間を低減する手段として、変更のあった部分のみを再度リンクするという部分リンク処理という技術が使われている。従来の部分リンク処理の一例が、特開平1-205332号公報に記載されている（従来技術）。

【0004】図13は、従来技術のリンク方法の一例を説明するためのフローチャートであり、図14、及び図15は、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図13のステップ1310、ステップ1311の各処理に対する配置後のメモリイメージである。

【0005】従来技術に記載された部分リンク及びリンク方法は、図13のように、結合部処理（ステップ1301）後に、まず配置対象の入力セグメントが存在するか判断する（ステップ1302）。ここで入力セグメントが存在しない場合は、配置処理の次の処理であるシンボル解決処理（ステップ1303）に移る。入力セグメントが存在する場合は、1回目のリンク処理か判断し（ステップ1304）、1回目リンク処理の場合は、対象セグメントを配置する（ステップ1305）。

【0006】1回目のリンク処理ではない場合（ステップ1304のNo）、セグメントが修正されていれば、そのセグメントのサイズを検出し、修正前のサイズより減少しているか同サイズであれば元の位置に配置（置換）する（ステップ1306のYes及び1308）。一方、修正されたセグメントのサイズが元のサイズよりも増加している場合は（ステップ1306のNo）、メモリ上で修正されたセグメントの終わりに空き領域があり、セグメント間の差分が空き領域に入る場合、元の位置に修正されたセグメントを置換配置する（ステップ1307のYes及び1308）。また、他のセグメントと重なる場合、メモリ全体で修正されたセグメントが配置できる空き領域が存在するか検出し（ステップ1309）、空き領域が存在する場合はその空き領域へ修正されたセグメントを移動配置し、元のセグメントは空き領域とする（ステップ1310、具体例は図14参照）。メモリ上に空き領域が存在しない場合は、全リンクにて再割付けを行う（ステップ1311、具体例は図15参照）。

【0007】

【発明が解決しようとする課題】一般的に、アプリケーションプログラム開発において、メモリ上の特定アドレスに配置しなければならないセグメントは、任意アドレスに配置可能なセグメントよりも少ないと考えられる。また、任意アドレスへ配置するセグメント間には、一般に空き領域が存在しない場合が多い。すなわち、本来、

リンク処理ではメモリを効率よく使うために、メモリ上に空き領域がないように配置するのが通常である。しかしながら、従来技術のリンク方法を実行した場合、修正されたセグメントのサイズが元のサイズよりも増加した場合、元の位置に配置できない割合が多くなり、メモリ上の他の空き領域への移動配置や全リンク処理を行うため、部分リンク処理に要する時間が大きくなってしまいう問題点があった。

【0008】本発明は斯かる問題点を鑑みてなされたものであり、その目的とするところは、修正されたセグメントのサイズが元のサイズよりも増加した場合であっても、元の位置に配置できない割合が多くなってメモリ上の他の空き領域への移動配置や全リンク処理を行うようなケースを回避し、リンク処理に要する時間を短縮できるリンク及びリンク方法を提供する点にある。

【0009】

【課題を解決するための手段】本発明の要旨は、複数のオブジェクト・モジュールをリンクして1つの実行可能プログラムを生成するリンク方法であって、オブジェクト・モジュールのプログラムの構成要素であるセグメントの中で配置対象となるセグメントを検出する手段と、当該検出された配置対象セグメント毎に、システム側で固定されて割り当てられた所定サイズの空き領域を割り当てる手段と、前記オブジェクト・モジュールのプログラムに変更があった前記配置対象セグメントのみを再度リンクする部分リンク処理を実行する手段と、前記部分リンク処理を実行する際、前記配置対象セグメント毎にシステム側で固定されて割り当てられた前記所定サイズの空き領域を当該対応する配置対象セグメントの終わりに各々付加する手段とを有することを特徴とするリンクに存する。また本発明の請求項2に記載の要旨は、前記配置対象セグメントがある場合、現在の前記部分リンク処理が1回目の部分リンクか、または前記部分リンク処理での全リンクかを判断する手段と、前記1回目の部分リンクまたは前記部分リンク処理での全リンクの場合、前記配置対象セグメントを配置する手段と、当該配置した配置対象セグメントの終わりに挿入する前記空き領域のサイズをシステム側で固定値に決定する手段と、当該固定の大きさの空き領域を当該対応する配置対象セグメントの終わりに各々挿入する手段とを有することを特徴とする請求項1に記載のリンクに存する。また本発明の請求項3に記載の要旨は、複数のオブジェクト・モジュールをリンクして1つの実行可能プログラムを生成するリンク方法であって、オブジェクト・モジュールのプログラムの構成要素であるセグメントの中で配置対象となるセグメントを検出する手段と、当該検出された配置対象セグメント毎に、ユーザ側が任意に割り当てた所定サイズの空き領域を割り当てる手段と、前記オブジェクト・モジュールのプログラムに変更があった前記配置対象セグメントのみを再度リンクする部分リンク処理を実行する手段

と、前記部分リンク処理を実行する際、前記配置対象セグメント毎にユーザ側が任意に割り当てた前記所定サイズの空き領域を当該対応する配置対象セグメントの終わりに各々付加する手段とを有することを特徴とするリンクに存する。また本発明の請求項4に記載の要旨は、前記配置対象セグメントがある場合、現在の前記部分リンク処理が1回目の部分リンクか、または前記部分リンク処理での全リンクかを判断する手段と、前記1回目の部分リンクまたは前記部分リンク処理での全リンクの場合、前記配置対象セグメントを配置する手段と、当該配置した配置対象セグメントの終わりに挿入する前記空き領域のサイズをユーザ側で任意の大きさに決定する手段と、当該任意の大きさの空き領域を当該対応する配置対象セグメントの終わりに各々挿入する手段とを有することを特徴とする請求項3に記載のリンクに存する。また本発明の請求項5に記載の要旨は、前記部分リンク処理の指示を入力する入力部と、前記配置対象セグメントの配置アドレスを決定する配置部と、前記配置部で決定した前記配置アドレスに基づいてリロケータブルなシンボルのシンボル値を解決するシンボル解決部と、前記リロケータブルなシンボルを参照している命令に基づいて解決した前記シンボル値を用いて命令コードの補正を行うコード補正部と、前記解決したコードを含む出力ファイルであるオブジェクトファイルを生成する出力部とを有することを特徴とする請求項1乃至4のいずれか一項に記載のリンクに存する。また本発明の請求項6に記載の要旨は、前記配置部は、一般的なリンクである全リンクを行う通常リンク部と、前記配置対象セグメントの終わりに前記空き領域を挿入するとともに、コード修正に応じたセグメントサイズの増減に対する前記部分リンク処理を行う部分リンク部を有することを特徴とする請求項5に記載のリンクに存する。また本発明の請求項7に記載の要旨は、複数のオブジェクト・モジュールをリンクして1つの実行可能プログラムを生成するリンク方法であって、オブジェクト・モジュールのプログラムの構成要素であるセグメントの中で配置対象となるセグメントを検出する工程と、当該検出された配置対象セグメント毎に、システム側で固定されて割り当てられた所定サイズの空き領域を割り当てる工程と、前記オブジェクト・モジュールのプログラムに変更があった前記配置対象セグメントのみを再度リンクする部分リンク処理を実行する工程と、前記部分リンク処理を実行する際、前記配置対象セグメント毎にシステム側で固定されて割り当てられた前記所定サイズの空き領域を当該対応する配置対象セグメントの終わりに各々付加する工程とを有することを特徴とするリンク方法に存する。また本発明の請求項8に記載の要旨は、前記配置対象セグメントがある場合、現在の前記部分リンク処理が1回目の部分リンクか、または前記部分リンク処理での全リンクかを判断する工程と、前記1回目の部分リンクまたは前記部分リン

ク処理での全リンクの場合、前記配置対象セグメントを配置する工程と、当該配置した配置対象セグメントの終わりに挿入する前記空き領域のサイズをシステム側で固定値に決定する工程と、当該固定の大きさの空き領域を当該対応する配置対象セグメントの終わりに各々挿入する工程とを有することを特徴とする請求項7に記載のリンク方法に存する。また本発明の請求項9に記載の要旨は、複数のオブジェクト・モジュールをリンクして1つの実行可能プログラムを生成するリンク方法であって、オブジェクト・モジュールのプログラムの構成要素であるセグメントの中で配置対象となるセグメントを検出する工程と、当該検出された配置対象セグメント毎に、ユーザ側が任意に割り当てた所定サイズの空き領域を割り当てる工程と、前記オブジェクト・モジュールのプログラムに変更があった前記配置対象セグメントのみを再度リンクする部分リンク処理を実行する工程と、前記部分リンク処理を実行する際、前記配置対象セグメント毎にユーザ側が任意に割り当てた前記所定サイズの空き領域を当該対応する配置対象セグメントの終わりに各々付加する工程とを有することを特徴とするリンク方法に存する。また本発明の請求項10に記載の要旨は、前記配置対象セグメントがある場合、現在の前記部分リンク処理が1回目の部分リンクか、または前記部分リンク処理での全リンクかを判断する工程と、前記1回目の部分リンクまたは前記部分リンク処理での全リンクの場合、前記配置対象セグメントを配置する工程と、当該配置した配置対象セグメントの終わりに挿入する前記空き領域のサイズをユーザ側で任意の大きさに決定する工程と、当該任意の大きさの空き領域を当該対応する配置対象セグメントの終わりに各々挿入する工程とを有することを特徴とする請求項9に記載のリンク方法に存する。

【0010】

【発明の実施の形態】以下、本発明の実施の形態を図面に基づいて詳細に説明する。

【0011】（第1実施形態）図1は、本発明のリンクの第1実施形態を説明するための機能ブロック図である。図1を参照すると、本実施形態のリンクは、入力部103、入力ファイル中の同名セグメントを1つにまとめる結合部104、セグメントの配置アドレスを決定する配置部105、シンボル解決部108、コード補正部109、出力部110を有する。

【0012】入力部103は、入力ファイルであるリロケータブル・オブジェクトファイル101、部分リンク処理を指示するオプション指定ファイル102とを処理する。シンボル解決部108は、配置部105で決定した配置アドレスに従いリロケータブルなシンボルのシンボル値を解決する。コード補正部109は、前記リロケータブルなシンボルを参照している命令に対し、解決したシンボル値を使い命令コードの補正を行う。出力部110は、解決したコードを含む出力ファイルであるオブ

ジェクトファイル111を生成する。

【0013】配置部105は、通常リンク部106と部分リンク部107とを備えている。通常リンク部106は、一般的なリンク（つまり、全リンク）を行う。部分リンク部107は、配置するセグメントの終わりに空き領域を挿入し、コード修正によるセグメントサイズの増減に対する部分リンク処理を行う。

【0014】次に図1、図2、図3、図4及び図5を参照して本実施形態の動作について詳細に説明する。配置部105は、リンク起動時の入力部103への指定に応じて、部分リンク処理が指定されたかを判断する。なお、通常リンク処理は本実施形態と直接関係ないので説明は割愛する。

【0015】図2は、図1のリンクで実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。部分リンク処理が選択された場合の配置処理では、図2に示すように、結合部処理（ステップ1301）後、まず配置対象の入力セグメントが存在するか判断する（ステップ1302）。

【0016】ここで入力セグメントが存在しない場合（ステップ1302のNo）、配置処理の次の処理であるシンボル解決処理（ステップ1303）を実行する。入力セグメントが存在する場合（ステップ1302のYes）、1回目の部分リンクであるか、または部分リンク処理での全リンクであるかを判断する（ステップ201）。

【0017】1回目の部分リンクまたは部分リンク処理での全リンクである場合（ステップ201のYes）、まず配置対象の当該セグメントを配置する（ステップ1305）。次にセグメントの終わりに挿入する空き領域のサイズをシステムで固定サイズに決定し（ステップ202）、そのサイズの空き領域を挿入する（ステップ203）。ここまでの処理により、メモリ上へ配置後のセグメント間には空き領域が存在することになる。

【0018】2回目以降の部分リンク処理の場合（ステップ201のNo）、修正されたセグメントのサイズが修正前のサイズより小さいかまたは同じ場合（ステップ1306のYes）、元の位置に配置する（ステップ1308）。元のサイズよりも大きい場合（ステップ1306のNo）、メモリ上で修正されたセグメントの直後に空き領域があり、そこに配置可能であるか判断する（ステップ1307）。

【0019】配置可能な場合（ステップ1307のYes）、元の位置に配置する（ステップ1308）。配置できない場合（ステップ1307のNo）、メモリ全体で修正されたセグメントを配置できる空き領域が存在するか判断する（ステップ1309）。

【0020】配置可能な空き領域が存在する場合（ステップ1309のYes）、その空き領域へ当該セグメン

トを配置し、元の領域は空き領域とする（ステップ1310）。空き領域が存在しない場合は（ステップ1309のNo）、メモリ上へ割付け不可能ということで、全リンク処理を行う（ステップ1311）。

【0021】図3は、図1のリンクで実行されるリンク方法であって、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図2のステップ1308の処理に対する配置後のメモリイメージである。修正されたセグメントのサイズが元のサイズ（リンク処理後のメモリイメージ中のB）よりも大きい場合（ステップ1306のNo）、メモリ上で修正されたセグメントの直後に空き領域（リンク処理後のメモリイメージ中のBの下に空き領域）があり、そこに配置可能であるか判断し（ステップ1307）、配置可能な場合（ステップ1307のYes）、元の位置（リンク処理後のメモリイメージ中のBの位置）に配置する（ステップ1308）。

【0022】図4は、図1のリンクで実行されるリンク方法であって、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図2のステップ1310の処理に対する配置後のメモリイメージである。メモリ上で修正されたセグメントの直後に空き領域があり、そこに配置可能であるか判断した結果（ステップ1307）、修正されたセグメントが配置できない場合（ステップ1307のNo）、メモリ全体で修正されたセグメントを配置できる空き領域が存在するか判断し（ステップ1309）、配置可能な空き領域が存在する場合（ステップ1309のYes）、その空き領域（メモリ再配置後のメモリイメージ中のBの領域）へ当該セグメントを配置し、元の領域は空き領域とする（ステップ1310）。

【0023】図5は、図1のリンクで実行されるリンク方法であって、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図2のステップ1311の処理に対する配置後のメモリイメージである。メモリ全体で修正されたセグメントを配置できる空き領域が存在するか判断した結果（ステップ1309）、空き領域が存在しない場合は（ステップ1309のNo）、メモリ上へ割付け不可能ということで、全リンク処理を行う（ステップ1311）。

【0024】以下に、ソースモジュールファイル数が10個、各ソースモジュールファイル中に記述されたセグメント数が10個のアプリケーションを例にとり効果を説明する。ただし、各セグメントの修正によるサイズ増分は、1とする。

【0025】本アプリケーションプログラムをデバッグ中の修正でコード増加するセグメント数は、全体の4割（40個）とする。従来の方法では、移動配置されるセグメント数は、このうち9割（36個）、空き領域がなく全リンクするのが1割（4seg=4個）とする。本

実施形態では、すべてのセグメント（40個）を元の位置に配置可能である。

【0026】フルリンクに要する処理時間を1とすると、元の位置に配置する処理時間は10分の1、移動配置する処理時間は10分の3程度と試算できる。これにより、上記例では、従来方法では14.8、本発明では4となり、本発明では従来よりも、リンク処理に要する時間をおおよそ1/3～1/4に減らすことができる。

【0027】一例として、約1Mバイトのアプリケーションプログラムに対し、上記処理を行った場合の実測の処理時間を示す。従来の方法では、約100秒ほどであるのに対し、本実施形態では、25秒ほどに削減することができた。なお、この実験では、CPUとしてPentium120MHz（米国インテル社製）を使用した。

【0028】以上第1実施形態を要約すれば、処理を記述したセグメントのサイズは大きい各セグメントの修正により増加するコードサイズは各セグメントのサイズと比較してかなり小さいものと考えられ、2回目以降の部分リンク時にセグメント間に空き領域が存在するため、セグメントのサイズが増加した場合でもセグメントの終わりに存在する空き領域を含めた元の位置に配置できる確率が上がり、処理時間を低減できる。移動配置する場合に必要であった元の領域をクリアする処理も、元の位置に配置できることで不要となる。また、全リンクする回数も減らすことができ、さらに処理時間を低減することが可能である。

【0029】（第2実施形態）次に、本発明の第2実施形態について図6及び図7を参照して説明する。図6は、本発明の第2実施形態のリンクで実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。なお、図6のうち第1実施形態の図2と同一の処理には同一番号を付し説明を省略する。なお、図6中の番号①の処理は図2中の番号①への分岐を示す。

【0030】図2に示された実施形態におけるシステムで固定の空き領域サイズを決定する処理（ステップ202）に対し、本実施形態では、セグメントの終わりに挿入する空き領域のサイズを決定する処理が、配置対象の当該セグメントのサイズの10%のサイズを算出する処理（ステップ601、602及び603）である点が図2に示された実施形態と異なる。

【0031】ステップ601で例えば10%のサイズとしたのは、アプリケーションプログラム開発において、デバッグ行程のコード修正で増加するセグメントサイズは経験的に7%程度という値が得られ、これにマージンを3%加味したためである。

【0032】次に、本実施形態の動作を図面を参照して詳細に説明する。部分リンク処理が選択された場合の配置処理では、図6示すように、結合部処理（ステップ1

301) 後、まず配置対象の入力セグメントが存在するか判断する(ステップ1302)。

【0033】ここで入力セグメントが存在しない場合(ステップ1302のNo)、配置処理の次の処理であるシンボル解決処理(ステップ1303)を実行する。入力セグメントが存在する場合(ステップ1302のYes)、1回目の部分リンクであるか、または部分リンク処理での全リンクであるかを判断する(ステップ201)。1回目の部分リンクまたは部分リンク処理での全リンクである場合(ステップ201のYes)、まず配置対象の当該セグメントを配置する(ステップ1305)。

【0034】図6のステップ601では、当該セグメントのサイズから、セグメントの終わりに挿入する空き領域のサイズとして、その10%のサイズを求める。

【0035】次に算出したサイズが1以上であるかを判断する(ステップ602)。サイズが1以上の場合(ステップ602のYes)、そのサイズの空き領域をステップ203で挿入する。サイズが1未満の場合(ステップ602のNo)、算出したサイズを1に補正し(ステップ603)、次のステップ203に進む。

【0036】図7は、図6の配置処理において、1回目の部分リンクまたは、部分リンク処理の全リンク処理でセグメント間に空き領域を挿入した配置後のメモリイメージである。すなわち、1回目の部分リンクまたは部分リンク処理での全リンクである場合(ステップ201のYes)、まず配置対象の当該セグメントA、Bを配置し(ステップ1305)、当該セグメントA、Bのサイズから、セグメントA、Bの終わりに挿入する空き領域のサイズとして、セグメントA、Bのサイズの10%のサイズ(=Aのサイズ/10、Bのサイズ/10)を求め(ステップ601)、算出したサイズが1以上であるかを判断した結果(ステップ602)、サイズが1以上の場合(ステップ602のYes)、そのサイズ(=Aのサイズ/10、Bのサイズ/10)の空き領域をステップ203で挿入(Aの下にAのサイズ/10の空き領域を挿入し、Bの下にBのサイズ/10の空き領域を挿入)する。

【0037】(第3実施形態)次に、本発明の第3実施形態について図8、図9及び図10を参照して説明する。図8は、本発明のリンクの第3実施形態を説明するための機能ブロック図である。図8を参照すると、図1に示された実施形態に対し、新たに空き領域サイズを指定するオプションファイル801を設けている。すなわち、図8を参照すると、本実施形態のリンクは、入力部103、入力ファイル中の同名セグメントを1つにまとめる結合部104、セグメントの配置アドレスを決定する配置部105、シンボル解決部108、コード補正部109、出力部110を有する。

【0038】前述したように、入力部103は、空き領域

サイズを指定するオプションファイル801、入力ファイルであるリロケータブル・オブジェクトファイル101、部分リンク処理を指示するオプション指定ファイル102を処理する。シンボル解決部108は、配置部105で決定した配置アドレスに従いリロケータブルなシンボルのシンボル値を解決する。コード補正部109は、前記リロケータブルなシンボルを参照している命令に対し、解決したシンボル値を使い命令コードの補正を行う。出力部110は、解決したコードを含む出力ファイルであるオブジェクトファイル111を生成する。配置部105は、通常リンク部106と部分リンク部107とを備えている。通常リンク部106は、一般的なリンク(つまり、全リンク)を行う。部分リンク部107は、配置するセグメントの終わりに空き領域を挿入し、コード修正によるセグメントサイズの増減に対する部分リンク処理を行う。

【0039】図9は、図8のリンクで実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。なお、図9の中で第1実施形態(図2)と同一の処理には同一番号を付し説明を省略する。なお、図9中の番号①の処理は、図2中の番号①への分岐を示す。図9を参照すると、本実施形態では、セグメントの終わりに挿入する空き領域のサイズを決定する処理が、図2に示された実施形態におけるシステムで固定の空き領域サイズを決定する処理(ステップ202)に対し、空き領域サイズを決定するためにユーザが前記空き領域サイズ指定オプションファイル801で指定した空き領域サイズを使用(ステップ901)する点が異なる。

【0040】本実施形態の動作を図9を参照して詳細に説明する。ユーザは、図8の空き領域サイズ指定オプションファイル801に、リンク起動時にサイズを指定する。

【0041】部分リンク処理が選択された場合の配置処理では、図9に示すように、結合部処理(ステップ1301)後、まず配置対象の入力セグメントが存在するか判断する(ステップ1302)。ここで入力セグメントが存在しない場合(ステップ1302のNo)、配置処理の次の処理であるシンボル解決処理(ステップ1303)を実行する。入力セグメントが存在する場合(ステップ1302のYes)、1回目の部分リンクであるか、または部分リンク処理での全リンクであるかを判断する(ステップ201)。1回目の部分リンクまたは部分リンク処理での全リンクである場合(ステップ201のYes)、まず配置対象の当該セグメントを配置する(ステップ1305)。

【0042】ステップ1305に続くステップ901では、空き領域サイズ指定オプションファイル801で指定されたサイズをセグメントの終わりに挿入する空き領域のサイズとして設定し、次のステップ203に進む。

ステップ203では、そのサイズの空き領域を挿入する。

【0043】以上説明したように、本実施形態のリンク方法は、第1実施形態の効果に加えて、アプリケーションプログラムの各処理のデバッグ状況を把握しているユーザが、現状のデバッグ状況に即した空き領域のサイズを指定できるため、最適な空き領域を空けることが可能になるといった特徴的な効果を備えている。

【0044】図10は、図9の配置処理において、1回目の部分リンクまたは、部分リンク処理の全リンク処理でセグメント間に空き領域を挿入した配置後のメモリイメージである。1回目の部分リンクであるか、または部分リンク処理での全リンクであるかを判断し（ステップ201）、1回目の部分リンクまたは部分リンク処理での全リンクである場合（ステップ201のYes）、まず配置対象の当該セグメントを配置し（ステップ1305）、空き領域サイズ指定オプションファイル801で指定されたサイズ（＝図8の空き領域サイズ指定オプションファイル801に、リンク起動時にユーザが指定したサイズ）をセグメントの終わりに挿入する空き領域のサイズとして設定し（ステップ901）、そのサイズの空き領域を挿入する（ステップ203）。

【0045】（第4実施形態）次に、本発明の第4の実施形態について図11及び図12を参照して説明する。図11は、本発明の第4実施形態のリンクで実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。なお、図11では、第1実施形態の図2と同一の処理には同一番号を付し説明を省略する。なお、図11中の番号①の処理は、図2中の番号①への分岐を示す。図11を参照すると、本実施形態では、セグメントの終わりに挿入する空き領域のサイズを決定する処理が、図2に示された実施形態におけるシステムで固定の空き領域サイズを決定する処理（ステップ202）に対し、メモリ上の全空き領域を全配置対象セグメントのサイズ比で分配して空き領域サイズを決定する処理（ステップ1101、1102、1103、1104、1105及び1106）である点が異なる。

【0046】本実施形態の動作を図面を参照して詳細に説明する。部分リンク処理が選択された場合の配置処理では、図11に示すように、結合部処理（ステップ1301）後、まず配置対象の入力セグメントが存在するか判断する（ステップ1302）。ここで入力セグメントが存在しない場合（ステップ1302のNo）、配置処理の次の処理であるシンボル解決処理（ステップ1303）を実行する。入力セグメントが存在する場合（ステップ1302のYes）、1回目の部分リンクであるか、または部分リンク処理での全リンクであるかを判断する（ステップ201）。1回目の部分リンクまたは部分リンク処理での全リンクである場合（ステップ201

のYes）、まず配置対象の当該セグメントを配置する（ステップ1305）。

【0047】図11のステップ1101では、最初の入力セグメントであるか判断する。最初の入力セグメントである場合（ステップ1101のYes）、配置対象セグメントの合計サイズを求める（ステップ1102）。次にステップ1102で求めた配置対象セグメントの合計サイズから、それら配置対象セグメントをメモリ上にすべて配置したときのメモリ上の全空き領域のサイズを求める（ステップ1103）。さらに、全配置対象セグメントのサイズ比を求め（ステップ1104）、ステップ1103で求めた全空き領域のサイズをこのサイズ比に分配し、その情報を格納する（ステップ1105）。

【0048】一方、ステップ1101において最初の入力セグメントでない場合（ステップ1101のNo）、ステップ1105で格納した各セグメント毎の空き領域サイズ情報を参照して当該セグメントに対する情報を読み出し（ステップ1106）、セグメントの終わりに挿入する空き領域のサイズとして設定し、セグメントの終わりに空き領域を挿入する（ステップ203）。

【0049】図12は、図11の配置処理において、1回目の部分リンクまたは、部分リンク処理の全リンク処理でセグメントA、B、C間に空き領域を挿入した配置後のメモリイメージである。

【0050】具体的には、セグメントAは最初の入力セグメントであるので（ステップ1101のYes）、配置対象セグメントA、B、Cの合計サイズ（ $A+B+C$ ）を求める（ステップ1102）。次にステップ1102で求めた配置対象セグメントA、B、Cの合計サイズ（ $A+B+C$ ）から、それら配置対象セグメントA、B、Cをメモリ上にすべて配置したときのメモリ上の全空き領域のサイズを求める（ステップ1103）。さらに、全配置対象セグメントA、B、Cのサイズ比（ $A/(A+B+C)$ 、 $B/(A+B+C)$ 、 $C/(A+B+C)$ ）を求め（ステップ1104）、ステップ1103で求めた全空き領域のサイズをこのサイズ比（ $A/(A+B+C)$ 、 $B/(A+B+C)$ 、 $C/(A+B+C)$ ）に分配し（全空き領域サイズ $\times A/(A+B+C)$ 、全空き領域サイズ $\times B/(A+B+C)$ 、全空き領域サイズ $\times C/(A+B+C)$ ）、その情報を格納し、（ステップ1105）セグメントの終わりに挿入する空き領域のサイズとして設定し、セグメントの終わりに空き領域を挿入する（ステップ203）。

【0051】以上説明したように、本実施形態のリンク方法では、第1実施形態の効果に加えて、メモリ全体の空き領域を各セグメント毎に分配するため、一番有効に空き領域を使用することができるといった特徴的な効果を備えている。

【0052】なお、上記構成部材の数、位置、形状等は上記実施の形態に限定されず、本発明を実施する上で好

適な数、位置、形状等にすることができる。また、各図において、同一構成要素には同一符号を付している。

【0053】

【発明の効果】本発明は、リンク処理時間を低減する手段として、変更のあった部分のみを再度リンクする部分リンク処理を実行する場合、プログラムの構成要素であるセグメントをメモリ上へ配置するとき、セグメント間に任意のサイズの空き領域を意図的に挿入することにより、部分リンク処理に要する処理時間を低減することが可能となるといった効果を奏する。

【図面の簡単な説明】

【図1】本発明のリンクの第1実施形態を説明するための機能ブロック図である。

【図2】図1のリンクで実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。

【図3】図1のリンクで実行されるリンク方法であって、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図2のステップ1308の処理に対する配置後のメモリイメージである。

【図4】図1のリンクで実行されるリンク方法であって、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図2のステップ1310の処理に対する配置後のメモリイメージである。

【図5】図1のリンクで実行されるリンク方法であって、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図2のステップ1311の処理に対する配置後のメモリイメージである。

【図6】本発明の第2実施形態のリンクで実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。

【図7】図6の配置処理において、1回目の部分リンクまたは、部分リンク処理の全リンク処理でセグメント間に空き領域を挿入した配置後のメモリイメージである。

【図8】本発明のリンクの第3実施形態を説明するための機能ブロック図である。

【図9】図8のリンクで実行されるリンク方法であっ

て、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。

【図10】図9の配置処理において、1回目の部分リンクまたは、部分リンク処理の全リンク処理でセグメント間に空き領域を挿入した配置後のメモリイメージである。

【図11】本発明の第4実施形態のリンクで実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。

【図12】図11の配置処理において、1回目の部分リンクまたは、部分リンク処理の全リンク処理でセグメント間に空き領域を挿入した配置後のメモリイメージである。

【図13】従来技術のリンク方法の一例を説明するためのフローチャートである。

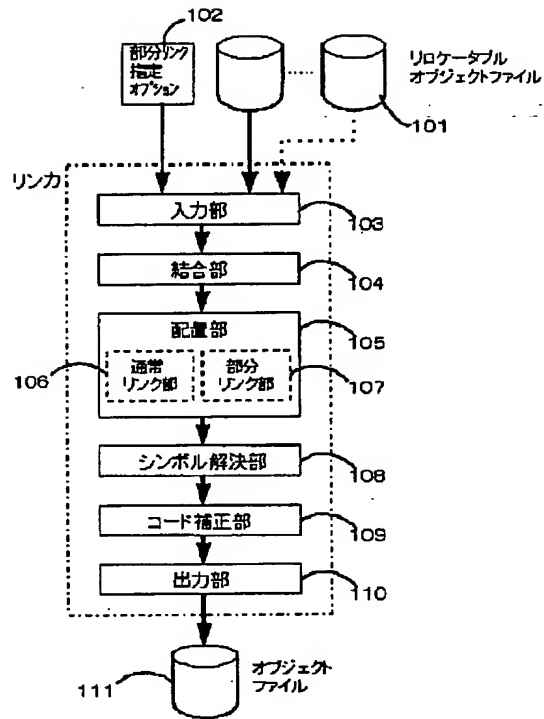
【図14】図13のリンク方法において、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図13のステップ1310の処理に対する配置後のメモリイメージである。

【図15】図13のリンク方法において、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図13のステップ1311の処理に対する配置後のメモリイメージである。

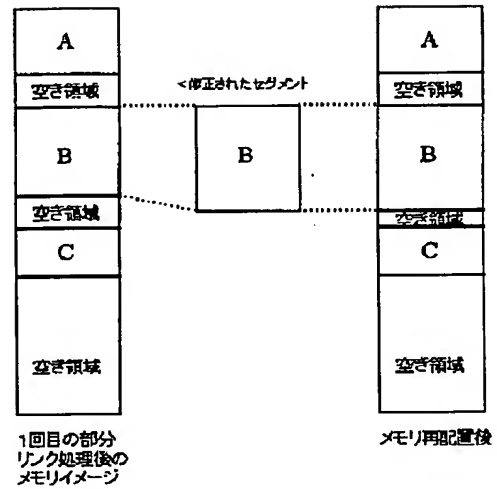
【符号の説明】

101…リロケータブル・オブジェクトファイル
102…オプション指定ファイル
103…入力部
104…結合部
105…配置部
106…通常リンク部
107…部分リンク部
108…シンボル解決部
109…コード補正部
110…出力部
111…オブジェクトファイル
801…オプションファイル

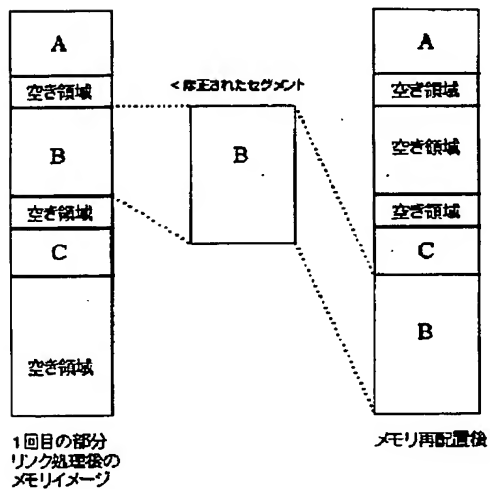
【図1】



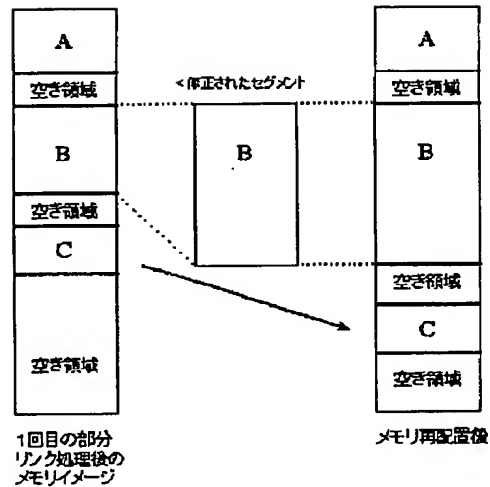
【図3】



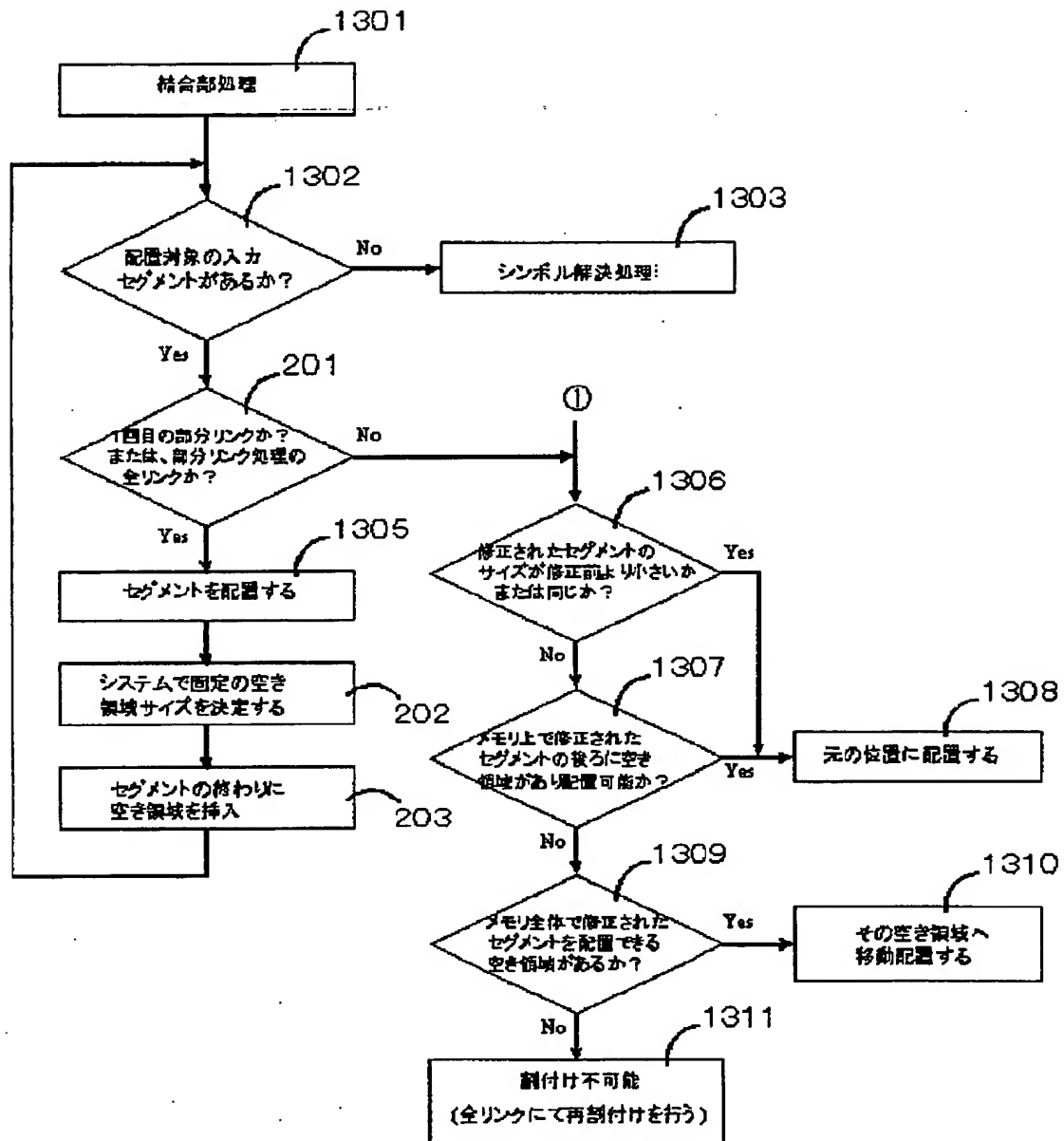
【図4】



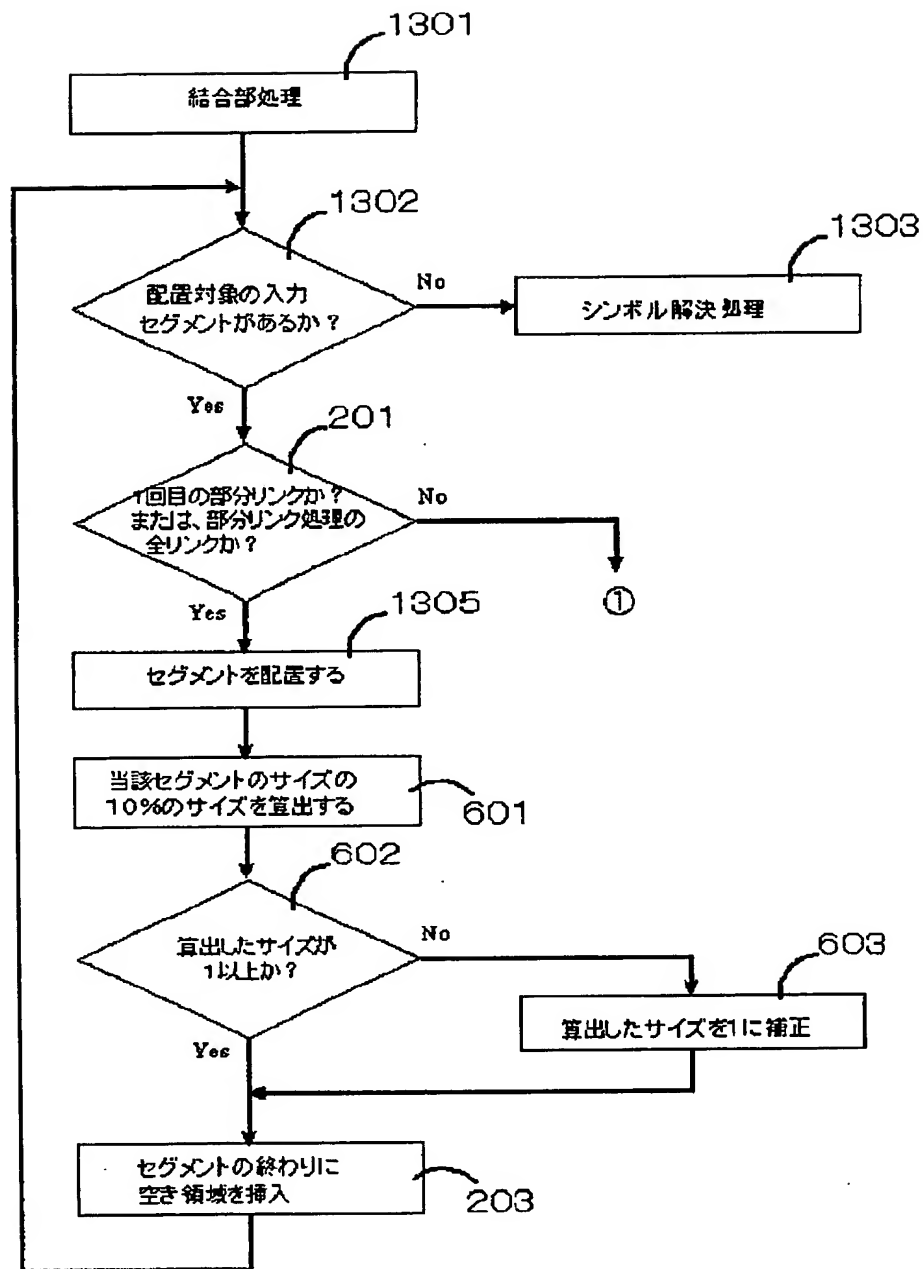
【図5】



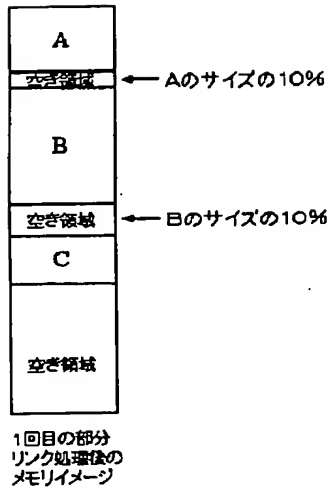
【図2】



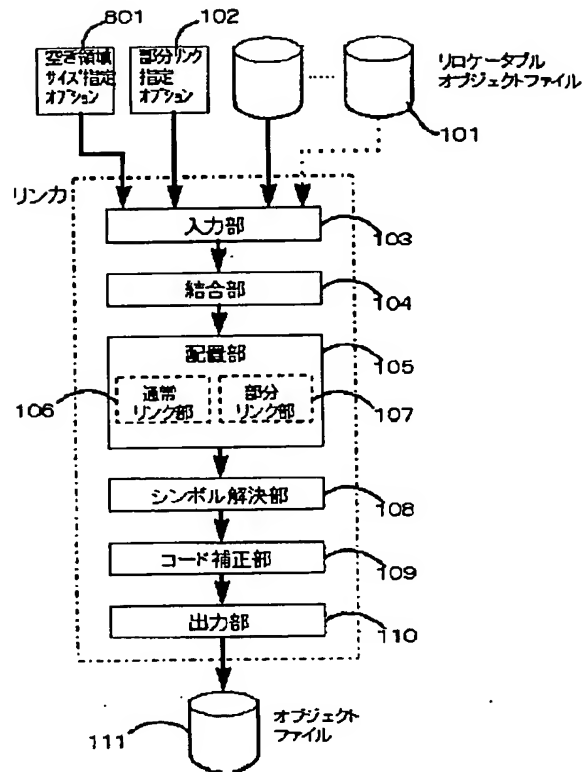
【図6】



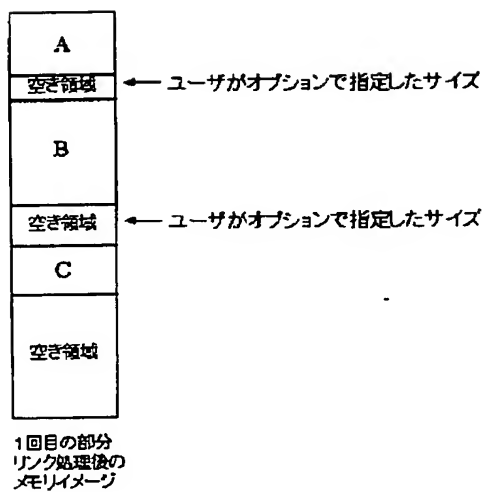
【図7】



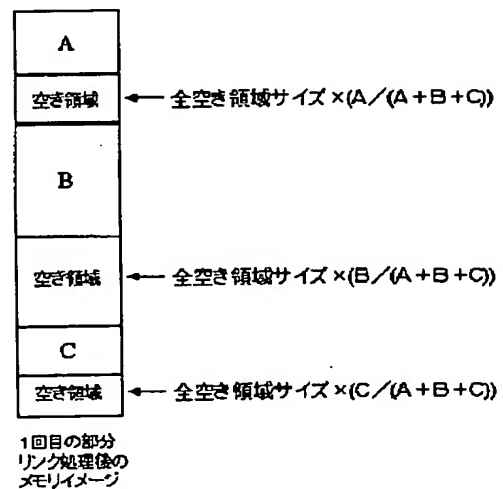
【図8】



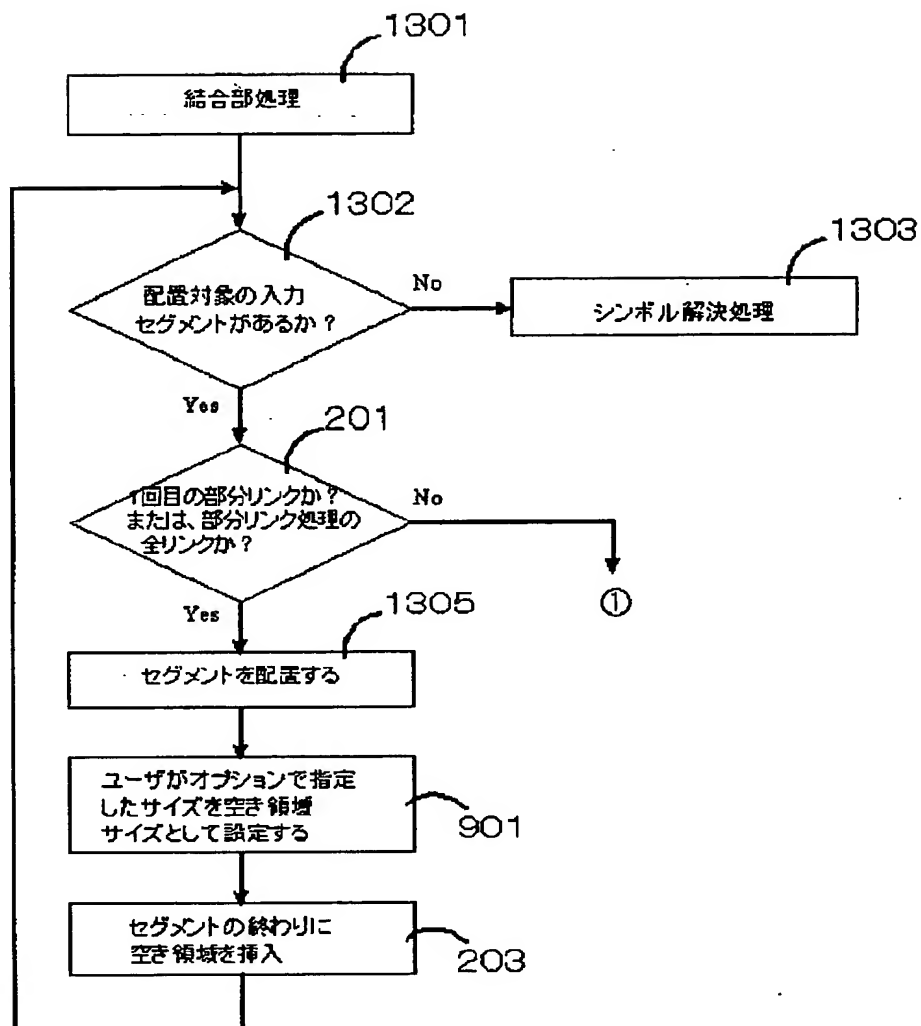
【図10】



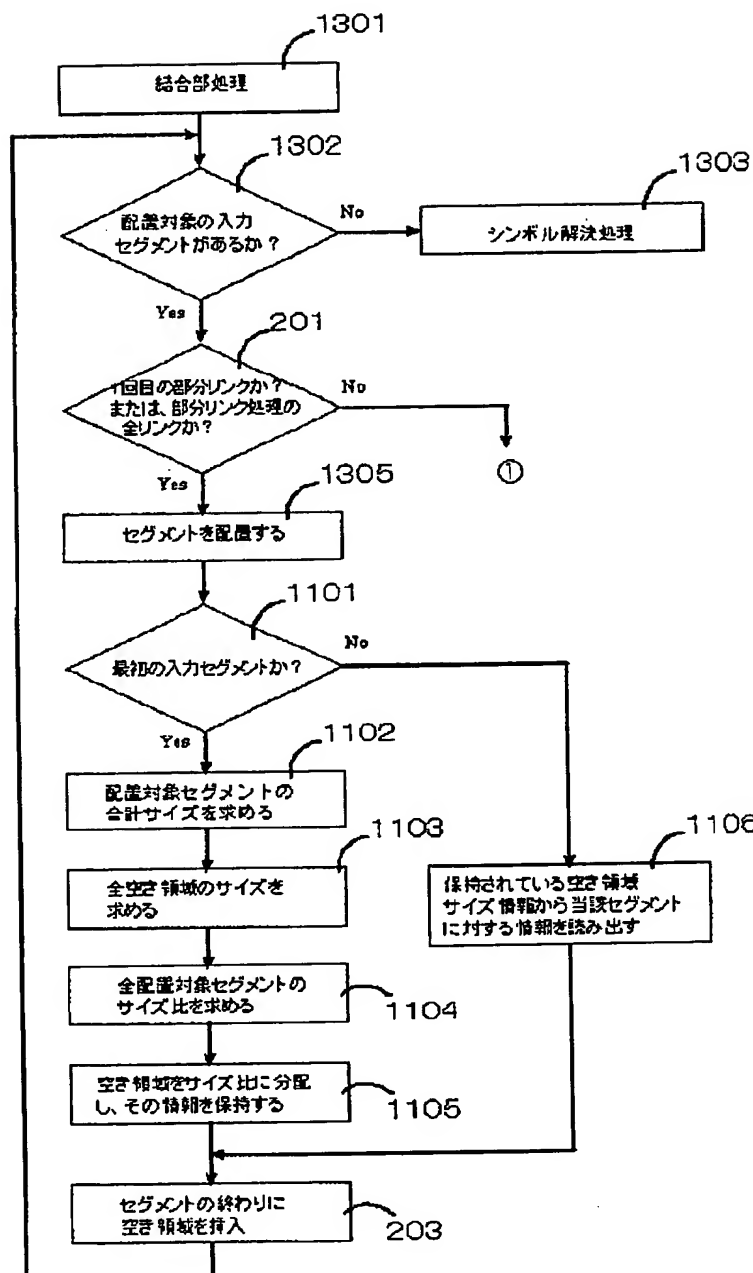
【図12】



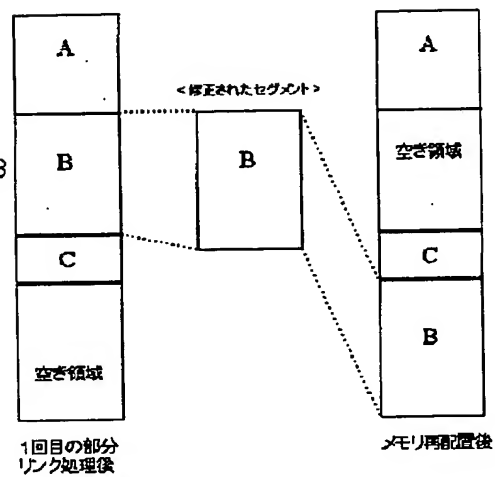
【図9】



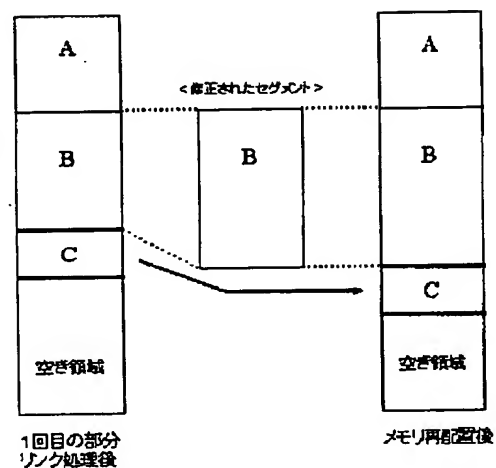
【図11】



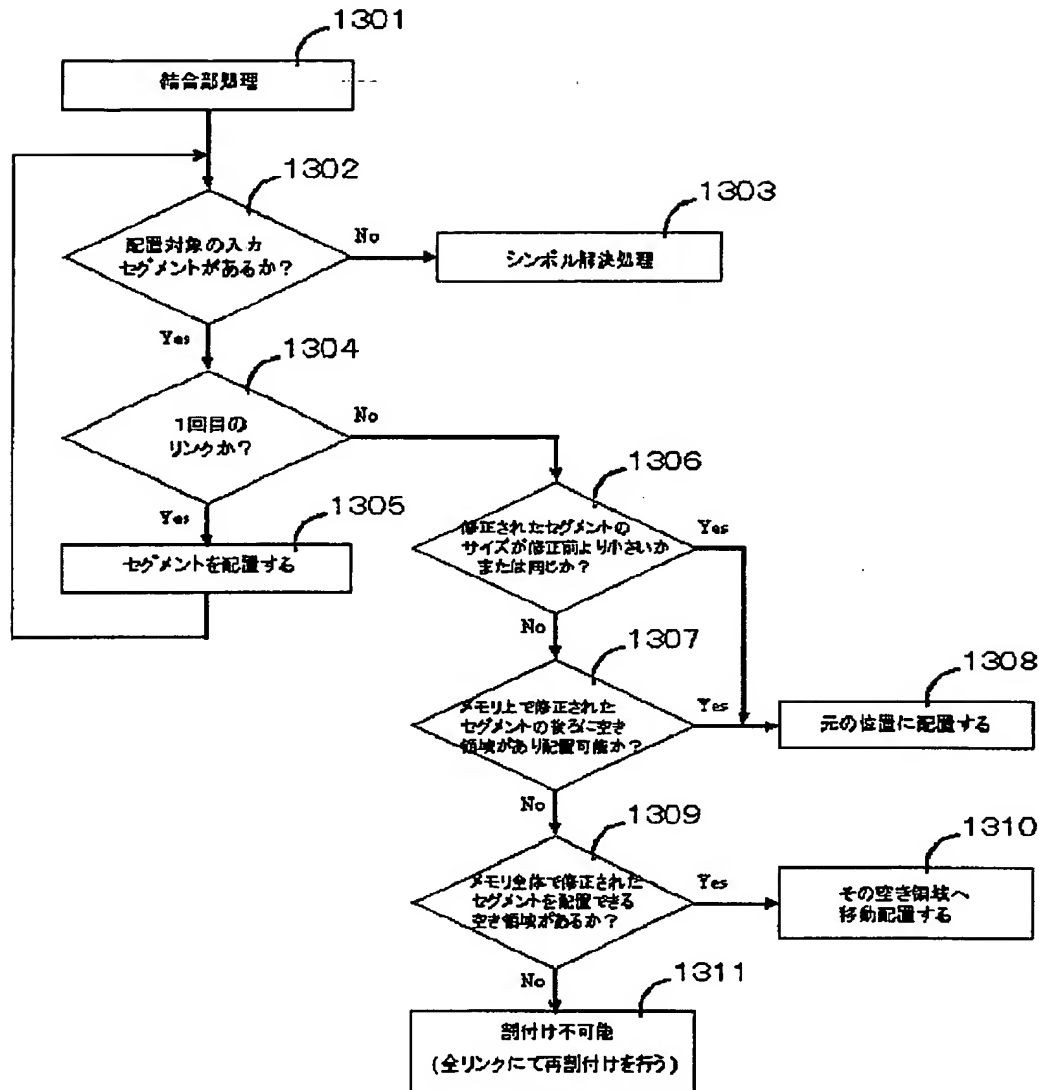
【図14】



【図15】



【図13】



【手続補正書】

【提出日】平成11年10月12日(1999.10.12)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】全文

【補正方法】変更

【補正内容】

【書類名】明細書

【発明の名称】リンク装置及びリンク方法

【特許請求の範囲】

【請求項1】 複数のオブジェクト・モジュールをリンクして1つの実行可能プログラムを生成するリンク装置であって、
各種ファイルを入力処理する入力部と、
入力ファイル中同名セグメントを一つにまとめる結合部と、
前記オブジェクト・モジュールのプログラムの構成要素であるセグメントの中で配置対象となるセグメントを検出し、全リンクを行う通常リンク部、および前記オブジェクト・モジュールのプログラムに変更があった前記セ

グメントの終わりに空き領域を挿入し部分リンク処理を行う部分リンク部の2つの部分を備えた配置部と、前記配置部で決定した配置アドレスに従いリロケータブルなシンボルのシンボル値を解決するシンボル解決部と、

前記リロケータブルなシンボルを参照している命令に対し、解決したシンボル値を使い命令コードの補正を行うコード補正部と、

オブジェクトファイルを生成する出力部を有することを特徴とするリンク装置。

【請求項2】 前記部分リンク部は、配置対象セグメントがある場合の部分リンク処理が1回目の部分リンクかまたは前記部分リンク処理での全リンクかを判断する手段を有することを特徴とする請求項1に記載のリンク装置。

【請求項3】 前記部分リンク部は、配置対象セグメントの終わりに挿入する前記空き領域のサイズをシステム側で固定値に決定する手段を有することを特徴とする請求項1または2に記載のリンク装置。

【請求項4】 前記部分リンク部は、配置対象セグメントの終わりに挿入する前記空き領域のサイズを配置対象セグメントサイズに一定のパーセンテージを掛け合わせたサイズに決定する手段を有することを特徴とする請求項1または2に記載のリンク装置。

【請求項5】 前記部分リンク部は、配置対象セグメントの終わりに挿入する前記空き領域のサイズをユーザ側で任意の大きさに決定する手段を有することを特徴とする請求項1または2に記載のリンク装置。

【請求項6】 前記部分リンク部は、配置対象セグメントの終わりに挿入する前記空き領域のサイズを配置対象セグメントの合計サイズ及び全空き領域のサイズ及び前記配置対象セグメントのサイズ比を求める手段と空き領域を前記サイズ比に分配する手段を有することを特徴とする請求項1または2に記載のリンク装置。

【請求項7】 複数のオブジェクト・モジュールをリンクして1つの実行可能プログラムを生成するリンク方法であって、

各種ファイルを入力処理する入力工程と、
入力ファイル中同名セグメントを一つにまとめる結合工程と、

前記オブジェクト・モジュールのプログラムの構成要素であるセグメントの中で配置対象となるセグメントを検出し、全リンクを行う通常リンク工程、および前記オブジェクト・モジュールのプログラムに変更があった前記セグメントの終わりに空き領域を挿入し部分リンク処理を行う部分リンク工程の2つの工程からなる配置工程と、

前記配置工程で決定した配置アドレスに従いリロケータブルなシンボルのシンボル値を解決するシンボル解決工程と、

前記リロケータブルなシンボルを参照している命令に対し、解決したシンボル値を使い命令コードの補正を行うコード補正工程と、

オブジェクトファイルを生成する出力工程を有することを特徴とするリンク方法。

【請求項8】 前記部分リンク工程は、配置対象セグメントがある場合の部分リンク処理が1回目の部分リンクかまたは前記部分リンク処理での全リンクかを判断する工程を有することを特徴とする請求項7に記載のリンク方法。

【請求項9】 前記部分リンク工程は、配置対象セグメントの終わりに挿入する前記空き領域のサイズをシステム側で固定値に決定する工程を有することを特徴とする請求項7または8に記載のリンク方法。

【請求項10】 前記部分リンク工程は、配置対象セグメントの終わりに挿入する前記空き領域のサイズを配置対象セグメントサイズに一定のパーセンテージを掛け合わせたサイズに決定する工程を有することを特徴とする請求項7または8に記載のリンク方法。

【請求項11】 前記部分リンク工程は、配置対象セグメントの終わりに挿入する前記空き領域のサイズをユーザ側で任意の大きさに決定する工程を有することを特徴とする請求項7または8に記載のリンク方法。

【請求項12】 前記部分リンク工程は、配置対象セグメントの終わりに挿入する前記空き領域のサイズを配置対象セグメントの合計サイズ及び全空き領域のサイズ及び前記両サイズの比を求める工程と空き領域をサイズ比に分配する工程を有することを特徴とする請求項7または8に記載のリンク方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、リンク技術に関し、特に、複数のオブジェクト・モジュールを結合（リンク）して1つの実行可能プログラムを生成するリンク装置及びリンク方法に関する。

【0002】

【従来の技術】近年のアプリケーションプログラム開発において、プログラム規模は増大し、より複雑なものになっている。一般的にアプリケーションプログラムのデバッグ工程では、コンパイル（アセンブル言語の場合はアセンブル）→リンク→デバッグ→コード修正という一連の作業を繰り返す。アプリケーションプログラムの規模増大により、上記の各作業に要する時間が大きくなっている。

【0003】リンク処理時間を低減する手段として、変更のあった部分のみを再度リンクする部分リンク処理という技術が使われている。従来の部分リンク処理の一例が、特開平1-205332号公報に記載されている（従来技術）。

【0004】図13は、従来技術のリンク方法の一例を

説明するためのフローチャートであり、図14、及び図15は、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図13のステップ1310、ステップ1311の各処理に対する配置後のメモリイメージである。

【0005】従来技術に記載された部分リンク装置及びリンク方法は、図13のように、結合部処理（ステップ1301）後に、まず配置対象の入力セグメントが存在するか判断する（ステップ1302）。ここで入力セグメントが存在しない場合は、配置処理の次の処理であるシンボル解決処理（ステップ1303）に移る。入力セグメントが存在する場合は、1回目のリンク処理か判断し（ステップ1304）、1回目リンク処理の場合は、対象セグメントを配置する（ステップ1305）。

【0006】1回目のリンク処理ではない場合（ステップ1304のNo）、セグメントが修正されていれば、そのセグメントのサイズを検出し、修正前のサイズより減少しているか同サイズであれば元の位置に配置（置換）する（ステップ1306のYes及び1308）。一方、修正されたセグメントのサイズが元のサイズよりも増加している場合は（ステップ1306のNo）、メモリ上で修正されたセグメントの終わりに空き領域があり、セグメント間の差分が空き領域に入る場合、元の位置に修正されたセグメントを置換配置する（ステップ1307のYes及び1308）。また、他のセグメントと重なる場合、メモリ全体で修正されたセグメントが配置できる空き領域が存在するか検出し（ステップ1309）、空き領域が存在する場合はその空き領域へ修正されたセグメントを移動配置し、元のセグメントは空き領域とする（ステップ1310、具体例は図14参照）。メモリ上に空き領域が存在しない場合は、全リンクにて再割付けを行う（ステップ1311、具体例は図15参照）。

【0007】

【発明が解決しようとする課題】一般的に、アプリケーションプログラム開発において、メモリ上の特定アドレスに配置しなければならないセグメントは、任意アドレスに配置可能なセグメントよりも少ないと考えられる。また、任意アドレスへ配置するセグメント間には、一般に空き領域が存在しない場合が多い。すなわち、本来、リンク処理ではメモリを効率よく使うために、メモリ上に空き領域がないように配置するのが通常である。しかしながら、従来技術のリンク方法を実行した場合、修正されたセグメントのサイズが元のサイズよりも増加した場合、元の位置に配置できない割合が多くなり、メモリ上の他の空き領域への移動配置や全リンク処理を行うため、部分リンク処理に要する時間が大きくなってしまいうという問題点があった。

【0008】本発明は斯かる問題点を鑑みてなされたものであり、その目的とするところは、修正されたセグ

メントのサイズが元のサイズよりも増加した場合であっても、元の位置に配置できない割合が多くなってメモリ上の他の空き領域への移動配置や全リンク処理を行うようなケースを回避し、リンク処理に要する時間を短縮できるリンク装置及びリンク方法を提供する点にある。

【0009】

【課題を解決するための手段】請求項1に記載の発明の要旨は、複数のオブジェクト・モジュールをリンクして1つの実行可能プログラムを生成するリンク装置であって、各種ファイルを入力処理する入力部と、入力ファイル中同名セグメントを一つにまとめる結合部と、前記オブジェクト・モジュールのプログラムの構成要素であるセグメントの中で配置対象となるセグメントを検出し、全リンクを行う通常リンク部、および前記オブジェクト・モジュールのプログラムに変更があった前記セグメントの終わりに空き領域を挿入し部分リンク処理を行う部分リンク部の2つの部分を備えた配置部と、前記配置部で決定した配置アドレスに従いロケータブルなシンボルのシンボル値を解決するシンボル解決部と、前記ロケータブルなシンボルを参照している命令に対し、解決したシンボル値を使い命令コードの補正を行うコード補正部と、オブジェクトファイルを生成する出力部を有することを特徴とするリンク装置に存する。また、請求項2に記載の発明の要旨は、前記部分リンク部は、配置対象セグメントがある場合の部分リンク処理が1回目の部分リンクかまたは前記部分リンク処理での全リンクかを判断する手段を有することを特徴とする請求項1に記載のリンク装置に存する。また、請求項3に記載の発明の要旨は、前記部分リンク部は、配置対象セグメントの終わりに挿入する前記空き領域のサイズをシステム側で固定値に決定する手段を有することを特徴とする請求項1または2に記載のリンク装置に存する。また、請求項4に記載の発明の要旨は、前記部分リンク部は、配置対象セグメントの終わりに挿入する前記空き領域のサイズを配置対象セグメントサイズに一定のパーセンテージを掛け合わせたサイズに決定する手段を有することを特徴とする請求項1または2に記載のリンク装置に存する。また、請求項5に記載の発明の要旨は、前記部分リンク部は、配置対象セグメントの終わりに挿入する前記空き領域のサイズをシステム側で任意の大きさに決定する手段を有することを特徴とする請求項1または2に記載のリンク装置に存する。また、請求項6に記載の発明の要旨は、前記部分リンク部は、配置対象セグメントの終わりに挿入する前記空き領域のサイズを配置対象セグメントの合計サイズ及び全空き領域のサイズ及び前記配置対象セグメントのサイズ比を求める手段と空き領域を前記サイズ比に分配する手段を有することを特徴とする請求項1または2に記載のリンク装置に存する。また、請求項7に記載の発明の要旨は、複数のオブジェクト・モジュールをリンクして1つの実行可能プログラムを生成する

リンク方法であって、各種ファイルを入力処理する入力工程と、入力ファイル中同名セグメントを一つにまとめる結合工程と、前記オブジェクト・モジュールのプログラムの構成要素であるセグメントの中で配置対象となるセグメントを検出し、全リンクを行う通常リンク工程、および前記オブジェクト・モジュールのプログラムに変更があった前記セグメントの終わりに空き領域を挿入し部分リンク処理を行う部分リンク工程の2つの工程からなる配置工程と、前記配置工程で決定した配置アドレスに従いリロケータブルなシンボルのシンボル値を解決するシンボル解決工程と、前記リロケータブルなシンボルを参照している命令に対し、解決したシンボル値を使い命令コードの補正を行うコード補正工程と、オブジェクトファイルを生成する出力工程を有することを特徴とするリンク方法に存する。また、請求項8に記載の発明の要旨は、前記部分リンク工程は、配置対象セグメントがある場合の部分リンク処理が1回目の部分リンクかまたは前記部分リンク処理での全リンクかを判断する工程を有することを特徴とする請求項7に記載のリンク方法に存する。また、請求項9に記載の発明の要旨は、前記部分リンク工程は、配置対象セグメントの終わりに挿入する前記空き領域のサイズをシステム側で固定値に決定する工程を有することを特徴とする請求項7または8に記載のリンク方法に存する。また、請求項10に記載の発明の要旨は、前記部分リンク工程は、配置対象セグメントの終わりに挿入する前記空き領域のサイズを配置対象セグメントサイズに一定のパーセンテージを掛け合わせたサイズに決定する工程を有することを特徴とする請求項7または8に記載のリンク方法に存する。また、請求項11に記載の発明の要旨は、前記部分リンク工程は、配置対象セグメントの終わりに挿入する前記空き領域のサイズをユーザ側で任意の大きさに決定する工程を有することを特徴とする請求項7または8に記載のリンク方法に存する。また、請求項12に記載の発明の要旨は、前記部分リンク工程は、配置対象セグメントの終わりに挿入する前記空き領域のサイズを配置対象セグメントの合計サイズ及び全空き領域のサイズ及び前記両サイズの比を求める工程と空き領域をサイズ比に分配する工程を有することを特徴とする請求項7または8に記載のリンク方法に存する。

【0010】

【発明の実施の形態】以下、本発明の実施の形態を図面に基づいて詳細に説明する。

【0011】(第1実施形態)図1は、本発明のリンク装置の第1実施形態を説明するための機能ブロック図である。図1を参照すると、本実施形態のリンク装置は、入力部103、入力ファイル中の同名セグメントを一つにまとめる結合部104、セグメントの配置アドレスを決定する配置部105、シンボル解決部108、コード補正部109、出力部110を有する。

【0012】入力部103は、入力ファイルであるリロケータブル・オブジェクトファイル101、部分リンク処理を指示するオプション指定ファイル102とを処理する。シンボル解決部108は、配置部105で決定した配置アドレスに従いリロケータブルなシンボルのシンボル値を解決する。コード補正部109は、前記リロケータブルなシンボルを参照している命令に対し、解決したシンボル値を使い命令コードの補正を行う。出力部110は、解決したコードを含む出力ファイルであるオブジェクトファイル111を生成する。

【0013】配置部105は、通常リンク部106と部分リンク部107とを備えている。通常リンク部106は、一般的なリンク(つまり、全リンク)を行う。部分リンク部107は、配置するセグメントの終わりに空き領域を挿入し、コード修正によるセグメントサイズの増減に対する部分リンク処理を行う。

【0014】次に図1、図2、図3、図4及び図5を参照して本実施形態の動作について詳細に説明する。配置部105は、リンク装置起動時の入力部103への指定に応じて、部分リンク処理が指定されたかを判断する。なお、通常リンク処理は本実施形態と直接関係ないので説明は割愛する。

【0015】図2は、図1のリンク装置で実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。部分リンク処理が選択された場合の配置処理では、図2に示すように、結合部処理(ステップ1301)後、まず配置対象の入力セグメントが存在するか判断する(ステップ1302)。

【0016】ここで入力セグメントが存在しない場合(ステップ1302のNo)、配置処理の次の処理であるシンボル解決処理(ステップ1303)を実行する。入力セグメントが存在する場合(ステップ1302のYes)、1回目の部分リンクであるか、または部分リンク処理での全リンクであるかを判断する(ステップ201)。

【0017】1回目の部分リンクまたは部分リンク処理での全リンクである場合(ステップ201のYes)、まず配置対象の当該セグメントを配置する(ステップ1305)。次にセグメントの終わりに挿入する空き領域のサイズをシステムで固定サイズに決定し(ステップ202)、そのサイズの空き領域を挿入する(ステップ203)。ここまでの処理により、メモリ上へ配置後のセグメント間には空き領域が存在することになる。

【0018】2回目以降の部分リンク処理の場合(ステップ201のNo)、修正されたセグメントのサイズが修正前のサイズより小さいかまたは同じ場合(ステップ1306のYes)、元の位置に配置する(ステップ1308)。元のサイズよりも大きい場合(ステップ1306のNo)、メモリ上で修正されたセグメントの直後

に空き領域があり、そこに配置可能であるか判断する(ステップ1307)。

【0019】配置可能な場合(ステップ1307のYes)、元の位置に配置する(ステップ1308)。配置できない場合(ステップ1307のNo)、メモリ全体で修正されたセグメントを配置できる空き領域が存在するか判断する(ステップ1309)。

【0020】配置可能な空き領域が存在する場合(ステップ1309のYes)、その空き領域へ当該セグメントを配置し、元の領域は空き領域とする(ステップ1310)。空き領域が存在しない場合は(ステップ1309のNo)、メモリ上へ割付け不可能ということで、全リンク処理を行う(ステップ1311)。

【0021】図3は、図1のリンク装置で実行されるリンク方法であって、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図2のステップ1308の処理に対する配置後のメモリイメージである。修正されたセグメントのサイズが元のサイズ(リンク処理後のメモリイメージ中のB)よりも大きい場合(ステップ1306のNo)、メモリ上で修正されたセグメントの直後に空き領域(リンク処理後のメモリイメージ中のBの下)の空き領域があり、そこに配置可能であるか判断し(ステップ1307)、配置可能な場合(ステップ1307のYes)、元の位置(リンク処理後のメモリイメージ中のBの位置)に配置する(ステップ1308)。

【0022】図4は、図1のリンク装置で実行されるリンク方法であって、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図2のステップ1310の処理に対する配置後のメモリイメージである。メモリ上で修正されたセグメントの直後に空き領域があり、そこに配置可能であるか判断した結果(ステップ1307)、修正されたセグメントが配置できない場合(ステップ1307のNo)、メモリ全体で修正されたセグメントを配置できる空き領域が存在するか判断し(ステップ1309)、配置可能な空き領域が存在する場合(ステップ1309のYes)、その空き領域(メモリ再配置後のメモリイメージ中のBの領域)へ当該セグメントを配置し、元の領域は空き領域とする(ステップ1310)。

【0023】図5は、図1のリンク装置で実行されるリンク方法であって、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図2のステップ1311の処理に対する配置後のメモリイメージである。メモリ全体で修正されたセグメントを配置できる空き領域が存在するか判断した結果(ステップ1309)、空き領域が存在しない場合は(ステップ1309のNo)、メモリ上へ割付け不可能ということで、全リンク処理を行う(ステップ1311)。

【0024】以下に、ソースモジュールファイル数が1

0個、各ソースモジュールファイル中に記述されたセグメント数が10個のアプリケーションを例にとり効果を説明する。ただし、各セグメントの修正によるサイズ増分は、1とする。

【0025】本アプリケーションプログラムをデバッグ中の修正でコード増加するセグメント数は、全体の4割(40個)とする。従来の方法では、移動配置されるセグメント数は、このうち9割(36個)、空き領域がなく全リンクするのが1割(4seg=4個)とする。本実施形態では、すべてのセグメント(40個)を元の位置に配置可能である。

【0026】フルリンクに要する処理時間を1とすると、元の位置に配置する処理時間は10分の1、移動配置する処理時間は10分の3程度と試算できる。これにより、上記例では、従来方法では14.8、本発明では4となり、本発明では従来よりも、リンク処理に要する時間をおおよそ1/3〜1/4に減らすことができる。

【0027】一例として、約1Mバイトのアプリケーションプログラムに対し、上記処理を行った場合の実測の処理時間を示す。従来の方法では、約100秒ほどであるのに対し、本実施形態では、25秒ほどに削減することができた。なお、この実験では、CPUとしてPentium120MHz(米国インテル社製)を使用した。

【0028】以上第1実施形態を要約すれば、処理を記述したセグメントのサイズは大きい各セグメントの修正により増加するコードサイズは各セグメントのサイズと比較してかなり小さいものと考えられ、2回目以降の部分リンク時にセグメント間に空き領域が存在するため、セグメントのサイズが増加した場合でもセグメントの終わりに存在する空き領域を含めた元の位置に配置できる確率が上がり、処理時間を低減できる。移動配置する場合に必要であった元の領域をクリアする処理も、元の位置に配置できることで不要となる。また、全リンクする回数も減らすことができ、さらに処理時間を低減することが可能である。

【0029】(第2実施形態)次に、本発明の第2実施形態について図6及び図7を参照して説明する。図6は、本発明の第2実施形態のリンク装置で実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。なお、図6のうち第1実施形態の図2と同一の処理には同一番号を付し説明を省略する。なお、図6中の番号①の処理は図2中の番号①への分岐を示す。

【0030】図2に示された実施形態におけるシステムで固定の空き領域サイズを決定する処理(ステップ202)に対し、本実施形態では、セグメントの終わりに挿入する空き領域のサイズを決定する処理が、配置対象の当該セグメントのサイズの10%のサイズを算出する処理(ステップ601、602及び603)である点が図

2に示された実施形態と異なる。

【0031】ステップ601で例えば10%のサイズとしたのは、アプリケーションプログラム開発において、デバッグ工程のコード修正で増加するセグメントサイズは経験的に7%程度という値が得られ、これにマージンを3%加味したためである。

【0032】次に、本実施形態の動作を図面を参照して詳細に説明する。部分リンク処理が選択された場合の配置処理では、図6示すように、結合部処理（ステップ1301）後、まず配置対象の入力セグメントが存在するか判断する（ステップ1302）。

【0033】ここで入力セグメントが存在しない場合（ステップ1302のNo）、配置処理の次の処理であるシンボル解決処理（ステップ1303）を実行する。入力セグメントが存在する場合（ステップ1302のYes）、1回目の部分リンクであるか、または部分リンク処理での全リンクであるかを判断する（ステップ201）。1回目の部分リンクまたは部分リンク処理での全リンクである場合（ステップ201のYes）、まず配置対象の当該セグメントを配置する（ステップ1305）。

【0034】図6のステップ601では、当該セグメントのサイズから、セグメントの終わりに挿入する空き領域のサイズとして、その10%のサイズを求める。

【0035】次に算出したサイズが1以上であるかを判断する（ステップ602）。サイズが1以上の場合（ステップ602のYes）、そのサイズの空き領域をステップ203で挿入する。サイズが1未満の場合（ステップ602のNo）、算出したサイズを1に補正し（ステップ603）、次のステップ203に進む。

【0036】図7は、図6の配置処理において、1回目の部分リンクまたは、部分リンク処理の全リンク処理でセグメント間に空き領域を挿入した配置後のメモリイメージである。すなわち、1回目の部分リンクまたは部分リンク処理での全リンクである場合（ステップ201のYes）、まず配置対象の当該セグメントA、Bを配置し（ステップ1305）、当該セグメントA、Bのサイズから、セグメントA、Bの終わりに挿入する空き領域のサイズとして、セグメントA、Bのサイズの10%のサイズ（ $=Aのサイズ/10$ 、 $Bのサイズ/10$ ）を求め（ステップ601）、算出したサイズが1以上であるかを判断した結果（ステップ602）、サイズが1以上の場合（ステップ602のYes）、そのサイズ（ $=Aのサイズ/10$ 、 $Bのサイズ/10$ ）の空き領域をステップ203で挿入（Aの下にAのサイズ/10の空き領域を挿入し、Bの下にBのサイズ/10の空き領域を挿入）する。

【0037】（第3実施形態）次に、本発明の第3実施形態について図8、図9及び図10を参照して説明する。図8は、本発明のリンク装置の第3実施形態を説明

するための機能ブロック図である。図8を参照すると、図1に示された実施形態に対し、新たに空き領域サイズを指定するオプションファイル801を設けている。すなわち、図8を参照すると、本実施形態のリンク装置は、入力部103、入力ファイル中の同名セグメントを1つにまとめる結合部104、セグメントの配置アドレスを決定する配置部105、シンボル解決部108、コード補正部109、出力部110を有する。

【0038】前述したように、入力部103は、空き領域サイズを指定するオプションファイル801、入力ファイルであるリロケータブル・オブジェクトファイル101、部分リンク処理を指示するオプション指定ファイル102を処理する。シンボル解決部108は、配置部105で決定した配置アドレスに従いリロケータブルなシンボルのシンボル値を解決する。コード補正部109は、前記リロケータブルなシンボルを参照している命令に対し、解決したシンボル値を使い命令コードの補正を行う。出力部110は、解決したコードを含む出力ファイルであるオブジェクトファイル111を生成する。配置部105は、通常リンク部106と部分リンク部107とを備えている。通常リンク部106は、一般的なリンク（つまり、全リンク）を行う。部分リンク部107は、配置するセグメントの終わりに空き領域を挿入し、コード修正によるセグメントサイズの増減に対する部分リンク処理を行う。

【0039】図9は、図8のリンク装置で実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。なお、図9の中で第1実施形態（図2）と同一の処理には同一番号を付し説明を省略する。なお、図9中の番号①の処理は、図2中の番号①への分岐を示す。図9を参照すると、本実施形態では、セグメントの終わりに挿入する空き領域のサイズを決定する処理が、図2に示された実施形態におけるシステムで固定の空き領域サイズを決定する処理（ステップ202）に対し、空き領域サイズを決定するためにユーザが前記空き領域サイズ指定オプションファイル801で指定した空き領域サイズを使用（ステップ901）する点が異なる。

【0040】本実施形態の動作を図9を参照して詳細に説明する。ユーザは、図8の空き領域サイズ指定オプションファイル801に、リンク装置起動時にサイズを指定する。

【0041】部分リンク処理が選択された場合の配置処理では、図9に示すように、結合部処理（ステップ1301）後、まず配置対象の入力セグメントが存在するか判断する（ステップ1302）。ここで入力セグメントが存在しない場合（ステップ1302のNo）、配置処理の次の処理であるシンボル解決処理（ステップ1303）を実行する。入力セグメントが存在する場合（ステップ1302のYes）、1回目の部分リンクである

か、または部分リンク処理での全リンクであるかを判断する(ステップ201)。1回目の部分リンクまたは部分リンク処理での全リンクである場合(ステップ201のYes)、まず配置対象の当該セグメントを配置する(ステップ1305)。

【0042】ステップ1305に続くステップ901では、空き領域サイズ指定オプションファイル801で指定されたサイズをセグメントの終わりに挿入する空き領域のサイズとして設定し、次のステップ203に進む。ステップ203では、そのサイズの空き領域を挿入する。

【0043】以上説明したように、本実施形態のリンク方法は、第1実施形態の効果に加えて、アプリケーションプログラムの各処理のデバッグ状況を把握しているユーザが、現状のデバッグ状況に即した空き領域のサイズを指定できるため、最適な空き領域を空けることが可能になるといった特徴的な効果を備えている。

【0044】図10は、図9の配置処理において、1回目の部分リンクまたは、部分リンク処理の全リンク処理でセグメント間に空き領域を挿入した配置後のメモリイメージである。1回目の部分リンクであるか、または部分リンク処理での全リンクであるかを判断し(ステップ201)、1回目の部分リンクまたは部分リンク処理での全リンクである場合(ステップ201のYes)、まず配置対象の当該セグメントを配置し(ステップ1305)、空き領域サイズ指定オプションファイル801で指定されたサイズ(=図8の空き領域サイズ指定オプションファイル801に、リンク装置起動時にユーザが指定したサイズ)をセグメントの終わりに挿入する空き領域のサイズとして設定し(ステップ901)、そのサイズの空き領域を挿入する(ステップ203)。

【0045】(第4実施形態)次に、本発明の第4の実施形態について図11及び図12を参照して説明する。図11は、本発明の第4実施形態のリンク装置で実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。なお、図11では、第1実施形態の図2と同一の処理には同一番号を付し説明を省略する。なお、図11中の番号①の処理は、図2中の番号①への分岐を示す。図11を参照すると、本実施形態では、セグメントの終わりに挿入する空き領域のサイズを決定する処理が、図2に示された実施形態におけるシステムで固定の空き領域サイズを決定する処理(ステップ202)に対し、メモリ上の全空き領域を全配置対象セグメントのサイズ比で分配して空き領域サイズを決定する処理(ステップ1101、1102、1103、1104、1105及び1106)である点が異なる。

【0046】本実施形態の動作を図面を参照して詳細に説明する。部分リンク処理が選択された場合の配置処理では、図11に示すように、結合部処理(ステップ13

01)後、まず配置対象の入力セグメントが存在するか判断する(ステップ1302)。ここで入力セグメントが存在しない場合(ステップ1302のNo)、配置処理の次の処理であるシンボル解決処理(ステップ1303)を実行する。入力セグメントが存在する場合(ステップ1302のYes)、1回目の部分リンクであるか、または部分リンク処理での全リンクであるかを判断する(ステップ201)。1回目の部分リンクまたは部分リンク処理での全リンクである場合(ステップ201のYes)、まず配置対象の当該セグメントを配置する(ステップ1305)。

【0047】図11のステップ1101では、最初の入力セグメントであるかを判断する。最初の入力セグメントである場合(ステップ1101のYes)、配置対象セグメントの合計サイズを求める(ステップ1102)。次にステップ1102で求めた配置対象セグメントの合計サイズから、それら配置対象セグメントをメモリ上にすべて配置したときのメモリ上の全空き領域のサイズを求める(ステップ1103)。さらに、全配置対象セグメントのサイズ比を求め(ステップ1104)、ステップ1103で求めた全空き領域のサイズをこのサイズ比に分配し、その情報を格納する(ステップ1105)。

【0048】一方、ステップ1101において最初の入力セグメントでない場合(ステップ1101のNo)、ステップ1105で格納した各セグメント毎の空き領域サイズ情報を参照して当該セグメントに対する情報を読み出し(ステップ1106)、セグメントの終わりに挿入する空き領域のサイズとして設定し、セグメントの終わりに空き領域を挿入する(ステップ203)。

【0049】図12は、図11の配置処理において、1回目の部分リンクまたは、部分リンク処理の全リンク処理でセグメントA、B、C間に空き領域を挿入した配置後のメモリイメージである。

【0050】具体的には、セグメントAは最初の入力セグメントであるので(ステップ1101のYes)、配置対象セグメントA、B、Cの合計サイズ($A+B+C$)を求める(ステップ1102)。次にステップ1102で求めた配置対象セグメントA、B、Cの合計サイズ($A+B+C$)から、それら配置対象セグメントA、B、Cをメモリ上にすべて配置したときのメモリ上の全空き領域のサイズを求める(ステップ1103)。さらに、全配置対象セグメントA、B、Cのサイズ比($A/(A+B+C)$ 、 $B/(A+B+C)$ 、 $C/(A+B+C)$)を求め(ステップ1104)、ステップ1103で求めた全空き領域のサイズをこのサイズ比($A/(A+B+C)$ 、 $B/(A+B+C)$ 、 $C/(A+B+C)$)に分配し(全空き領域サイズ $\times A/(A+B+C)$ 、全空き領域サイズ $\times B/(A+B+C)$ 、全空き領域サイズ $\times C/(A+B+C)$)、その情報を格納し、(ステップ1105)セグメントの終わりに挿入す

る空き領域のサイズとして設定し、セグメントの終わりに空き領域を挿入する（ステップ203）。

【0051】以上説明したように、本実施形態のリンク方法では、第1実施形態の効果に加えて、メモリ全体の空き領域を各セグメント毎に分配するため、一番有効に空き領域を使用することができるといった特徴的な効果を備えている。

【0052】なお、上記構成部材の数、位置、形状等は上記実施の形態に限定されず、本発明を実施する上で好適な数、位置、形状等にすることができる。また、各図において、同一構成要素には同一符号を付している。

【0053】

【発明の効果】本発明は、リンク処理時間を低減する手段として、変更のあった部分のみを再度リンクする部分リンク処理を実行する場合、プログラムの構成要素であるセグメントをメモリ上へ配置するとき、セグメント間に任意のサイズの空き領域を意図的に挿入することにより、部分リンク処理に要する処理時間を低減することが可能となるといった効果を奏する。

【図面の簡単な説明】

【図1】本発明のリンク装置の第1実施形態を説明するための機能ブロック図である。

【図2】図1のリンク装置で実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。

【図3】図1のリンク装置で実行されるリンク方法であって、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図2のステップ1308の処理に対する配置後のメモリイメージである。

【図4】図1のリンク装置で実行されるリンク方法であって、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図2のステップ1310の処理に対する配置後のメモリイメージである。

【図5】図1のリンク装置で実行されるリンク方法であって、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図2のステップ1311の処理に対する配置後のメモリイメージである。

【図6】本発明の第2実施形態のリンク装置で実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。

【図7】図6の配置処理において、1回目の部分リンクまたは、部分リンク処理の全リンク処理でセグメント間に空き領域を挿入した配置後のメモリイメージである。

【図8】本発明のリンク装置の第3実施形態を説明するための機能ブロック図である。

【図9】図8のリンク装置で実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。

【図10】図9の配置処理において、1回目の部分リンクまたは、部分リンク処理の全リンク処理でセグメント

間に空き領域を挿入した配置後のメモリイメージである。

【図11】本発明の第4実施形態のリンク装置で実行されるリンク方法であって、部分リンク処理が選択された場合の配置処理を説明するためのフローチャートである。

【図12】図11の配置処理において、1回目の部分リンクまたは、部分リンク処理の全リンク処理でセグメント間に空き領域を挿入した配置後のメモリイメージである。

【図13】従来技術のリンク方法の一例を説明するためのフローチャートである。

【図14】図13のリンク方法において、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図13のステップ1310の処理に対する配置後のメモリイメージである。

【図15】図13のリンク方法において、修正されたセグメントのサイズが修正前より大きくなった場合に行われる図13のステップ1311の処理に対する配置後のメモリイメージである。

【符号の説明】

101…リロケートブル・オブジェクトファイル

102…オプション指定ファイル

103…入力部

104…結合部

105…配置部

106…通常リンク部

107…部分リンク部

108…シンボル解決部

109…コード補正部

110…出力部

111…オブジェクトファイル

801…オプションファイル

【手続補正2】

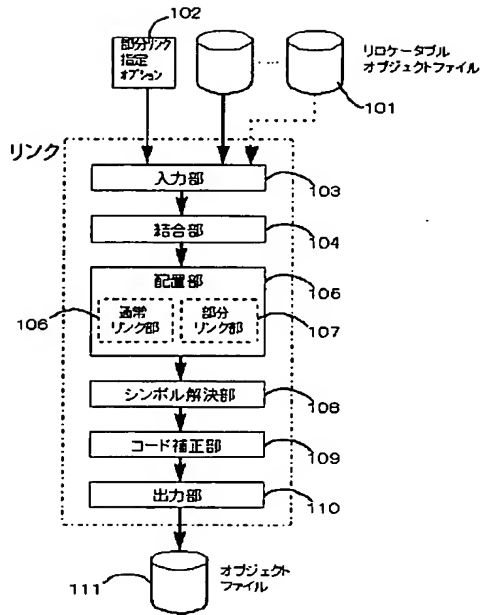
【補正対象書類名】図面

【補正対象項目名】図1

【補正方法】変更

【補正内容】

【図1】

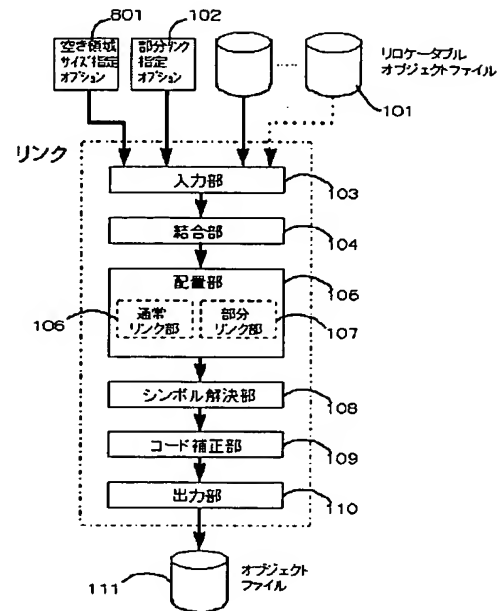


【手続補正3】
【補正対象書類名】図面
【補正対象項目名】図8

【補正方法】変更

【補正内容】

【図8】



フロントページの続き

(72)発明者 社本 英司
神奈川県川崎市中原区小杉町一丁目403番
53 日本電気アイシーマイコンシステム株
式会社内

Fターム(参考) 5B076 AB02